Requirements for Load Tests in the Integration Program

http://www.usatlas.bnl.gov/twiki/bin/view/Admins/LoadTestsP1

Version 2
June 12, 2007

Overview

We first describe a set of load test types that will be executed either autonomously at the local site level, or in a scheduled, coordinated fashion between sites.

Local access

A site admin is given a little script or toolkit which measures local I/O performance as would be seen by an ATLAS production or analysis job.

Example: running bonnie++ with suitable options to get benchmarks for sequential read and write operations, reported in a common format, between a host CPU and its attached, local scratch disks.

Variants could include I/O over NFS, PNFS (dCache), xrootd, or GPFS mounted volumes.

Single door load tests

A site admin uses a simple tool that stresses access to the sites SE in a few key ways:

- Simple read/writes of test datasets via Gridftp, establishing number of simultaneous transfers before exceeding a load of say 5, effective throughput by a single server. The datasets can be defined according to types: A) analysis-like – many files of moderate size (< 100 MB), versus B) production-like: fewer large files, but larger in size on average (> 100MB).
- A tool could be provided to generate these fake datasets according to named recipes.
- Similar exercises, but for employing the SRM door.
- Also, memory-to-memory to isolate the disks.
- Exercising the FTS layers as well.

These can be done locally from a Tier2 or Tier2 edge server, so as to separate local from wide-area network effects.

Multi-door load tests

Similar to single door load tests, but data is transferred through multiple doors to measure and effective site-level throughput scale.

Wide area data transfer load tests

A site admin is given a simple toolkit that has the ability to exercise single door load tests between two endpoints in the cloud.

Site-to-Site wide area data transfer load tests

Extend the simple point-to-point transfer scheme to using multiple servers at the source/destination and measure effective throughput. This again could be performed by the site system administrators using a load testing toolkit.

Persistent data transfer load tests

A managed operation out of the Tier1 most likely that conducts wide area data transfer or site-to-site load tests over a sustained time period, demonstrating robustness of operation.

Job robot load testing

A load test that exercises access to the local gatekeepers (GT2 or GT4 Gram) in a variety of job types:

- Type I: simple submit, no I/O involved. Scale.
- Type II: data-producing job: return data to user's submit host. Scale
- Type III: data-producing job: transfer data to some other designated storage area. Scale.
- Type IV: data-consuming job, transfer input data form users's submit host. Scale
- Type V: data-consuming job, transfer input data from some other designated storage area. Scale
- Type VI: data-consuming/data-producing job, to and from the submit host. Scale.
- Type VII: data-consuming/data-producing job, from arbitrary endpoints. Scale.

Metrics and Dashboard

We need to collect in one place a disparate set of sensor reports for the Facility – ideally we'd like to selectively pick out certain Ganglia type plots from each of the endpoints in the system so we can examine, record and archive loads, information about processes, etc. This would be a difficult project – it would have to be laid out and organized carefully, and we'd need buy-in from each of the sites. It could though turn out to make life quite a bit easier. Define metrics

For monitoring an assortment of metrics. See for example, https://www.racf.bnl.gov/Facility/Monitor/dashboard.html.

Scheduler

For scheduling multi-site transfers and other load tests via a web interface. It may be possible to reuse the Terapaths software for this purpose.

Tasks

1. Define specific load tests and the scripts/tools/services needed for each.
2. Setup repository and cache for load testing tools.
3. Initial set of load testing tools
    a. Create local access load tester scripts.
    b. Create single-door load tester scripts.
    c. Create multi-door load tester scripts.
    d. Create pair-wise site load tester scripts.
4. Implement a simple load testing dashboard consisting of load test configuration panels and response monitors
5. Implement control system for coordinated, multi-site load tests.